Immune Evolutionary Algorithms

WANG Lei and JIAO Licheng
Kay Lab for Radar Signal Processing, Xidian University, Xi'an, 710071
Email: *wanglei@rsp.xidian.edu.cn*

Abstract

Three novel evolutionary algorithms, the immune genetic algorithm (IGA), the immune evolutionary programming (IEP) and the immune evolutionary strategy (IES), are presented based on the immune theory in biology, which are not only convergent but used for solving complex discrete optimization problems as well. They all construct an immune operator accomplished by two components, a vaccination and an immune selection. The methods for selecting vaccines and constructing an immune operator are also proposed. Simulations show that these algorithms can restrain the degenerate phenomenon and improve the searching capability of the existing algorithms, therefore increase the convergent speed greatly.

Keywords

GA, EP, ES, immunity, convergence, simulations.

1. Introduction

During the last four decades, there has been a growing interest in algorithms that rely on analogies to natural phenomena such as evolution, heredity, immunity and so on. The emergence of massively parallel computers made these algorithms of practical interest. Evolutionary algorithms just belong to these best known algorithms, whose beginnings can be tracked back to the early 1950s when several biologists used computers for simulations of biological systems[1]. However, the work done in late 1960s and early 1970s at the University of Michigan under the direction of John Holland led to genetic algorithms (GA) as it is known today[2]. In the 1960s, Fogel and Owens et al used the related knowledge for reference and applied them into the field of engineering, and good results were received, then the evolutionary programming (EP) was formed[3]. Also in the 1960s, based on analyzing and utilizing evolutionism, Rechenberg and Schwefel et al used the genetic theory and the selection mechanism in the field of engineering, such as the problems like a flexible pipe bending or changeable nozzle contour into a shape with minimal loss of energy, therefore the evolutionary strategy (ES) appeared gradually[4]. With the characteristics of easier application, grater robustness and batter parallel processing than most classical methods of optimization, these algorithms have been widely used for combinatorial optimization[5][6], structural designing[7], rule-based machine learning systems[8][9] and other engineering problems[10]-[12]. In recent years, the researches on evolutionary algorithms mainly focus on dealing with the optimal problem shown as follows:

$$f^* := f(\vec{x}^*) = \max\left\{ f(\vec{x}) | \vec{x} \in \Omega \subseteq R^n \right\} \qquad (1)$$

where $\Omega = \left\{ \vec{x} \in R^n | g_j \geq 0 \quad \forall j \in \{1, \cdots, q\} \right\}$ is the set of all points, $g_j : R^n \to R$ is an inequality restriction and $f(\vec{a})$ is the fitness function of an individual $\vec{a}$.

Evolutionary algorithms are all good searching algorithms with an iterative process of generation-and-test. Two operators, crossover and mutation, give each individual the chance of optimization and ensure the evolutionary tendency with the select mechanism of survival of the fittest. However, the two operators change individuals randomly and indirectly under some conditions. Therefore, they not only give individuals the evolutionary chance but also cause certain degeneracy. On the other hand, there are some basic and obvious characteristics or knowledge in pending problems. However the operators are lack of the capability of meeting an actual situation, so appear some torpid when solving a problem, which is conducive to the universality of the algorithm but neglects the assistant function of the characteristics or knowledge. The loss due to this negligence is sometimes considerable in dealing with complex problems. It is also realized in the practice that only using evolutionary algorithms is deficient for simulating the ability with which the human beings settle problems. Therefore, it is a perpetual theme in the research field of evolutionary algorithms and even in the intelligent computation how to learn, exploit and utilize the human intelligence.

In the view of biology, evolution is a process of optimizing population with the selection mechanism. Immunity is the means by which the organism protects itself through the neutralization between antibodies and antigens. If an original evolutionary algorithm is considered as an organism, then the degeneracy rising inevitably from performing the algorithm can be regarded as an outside antigen. It can be just looked upon as an neutral course that the algorithm utilizes the characteristic information of the pending problem to restrain the above degeneracy through vaccination (an algorithmic approach). The vaccination takes effect of survival of the fittest on the original evolutionary algorithm, keeps its every operation in the optimal direction, and then realizes the goal of immunity.

Introducing immune concepts and methods into evolutionary algorithms is to seek the ways and means for utilizing local characteristic information to find the optimal solution when dealing with difficult problems. In some detail, each novel algorithm proposed in this paper intervenes in the globally parallel process with the local information, restrains or avoids repetitive and useless work. They overcome the blindness of the crossover and the mutation operation and make the fitness of population rise steadily. The above course is very similar with that of immune phenomenon in nature[13], the algorithms based on the above idea are so named immune evolutionary algorithms for the purpose of simplicity and direct-view. Immune evolutionary algorithms includes three ones, i.e., the immune genetic algorithm (IGA), the immune evolutionary programming (IEP) and the immune evolutionary strategy (IES).

2. The Immune Genetic Algorithm

In this algorithm, the idea of immunity is mainly realized through two steps based on reasonably selecting vaccines, i.e. a vaccination and an immune selection, of which the former is used for raising fitness and the latter is for preventing the deterioration.

2.1 Vaccination

A vaccination means the course of modifying the genes of an individual x on some bits in accordance with priori knowledge so as to gain higher fitness with greater probability. Suppose a population is $c=(x_1,x_2,\cdots,x_{n_0})$, and then the vaccination on c means the operation carried out on $n_\alpha=\alpha n$ individuals that are selected from c in proportion as α. A vaccine is abstracted form the prior knowledge of the pending problem, which the information contained in and the validity of take an important role in the performance of the algorithm.

2.2 Immune selection

This operation is accomplished by the following two components. The first one is the immune test, i.e. testing the antibodies. If the fitness is smaller than that of the parent, which means serious degeneration must have happened in the process of crossover or mutation, then the parent will attend the next competition instead of the individual. The second one is the annealing selection[14], i.e. selecting an individual x_i in the present offspring $E_k=(x_1\cdots x_{n_0})$ to join in the new parents with the probability as follows:

$$P(x_i)=\frac{e^{f(x_i)/T_k}}{\sum_{i=1}^{n_0} e^{f(x_i)/T_k}} \tag{2}$$

where $f(x_i)$ is the fitness of the individual x_i and $\{T_K\}$ is the temperature-controlled series tending towards 0.

2.3 IGA and Its Convergence

1. Create initial random population A_1.
2. Abstract vaccines according to the prior knowledge.
3. If the current population contains the optimal individual, then the course halts; else, continues.
4. Perform crossover on the k-th parent A_k and obtain the results B_k.
5. Perform mutation on B_k and obtain C_k.
6. Perform vaccination on C_k and obtain D_k.
7. Perform immune selection on D_k and obtain the next parent A_{k+1}, then go to 3.

Given the size of a population is n_0 and the encoding of an individual is the q-scale code with l-bit. The crossover is carried out with one point or multi-points. The mutation is performed on each gene bit independently, after which the probability of being any other state is $1/q-1$. Then the transformation of the states in the algorithm is shown as follows:

$$A_k \xrightarrow{\text{Crossover}} B_k \xrightarrow{\text{Mutation}} C_k \xrightarrow{\text{Vaccination}} D_k \xrightarrow{\text{Immunen Selection}} A_{k+1}$$

where the transformation of the states from Ak to Dk constitutes a Markov chain, while the state Ak+1 is related to each of the former states. However, the random process { Ak | k = 1, 2,$\cdots$} is still a Markov process. Suppose that X is a searching space, we consider the population with the size of n_0 as a point in the state space $S=X^{n_0}$, in which each coordinate is an individual in X. Suppose |S| indicates the number of the states in S; $s_i\in S$ (i=1, 2,$\cdots$, |S|) expresses s_i is a certain state in S; $s_i\subseteq s_j$ shows s_j contains s_i when they are all the subsets in X; V_k^i suggests that a random variable V is just in the state s_i of the k-th generation, and f is the fitness function on X. Let:

$$S^*=\{x\in X\,|\,f(x)=\max_{x_i\in X} f(x_i)\} \tag{3}$$

then the convergence of the algorithm can be defined as follows:

Definition 2.1: For any initial distribution, if the following equation holds,

$$\lim_{k\to\infty}\sum_{s_i\cap S^*\neq\emptyset} P\{A_k^i\}=1 \tag{4}$$

then the algorithm is convergent.

This definition of the convergence means that if the algorithm implements for enough iteration, then the probability with which the population contains the optimal individual will verge on 1. So the definition shown as the above is usually called the convergence with probability 1.

Theorem 2.1: The immune genetic algorithm is convergent with probability 1.

It is necessary to point out that if the immune operator is cut off from this algorithm, then it can be proved not to converge to the global optimal individual[15], or it is strongly non-convergent[14].

3. The Immune Evolutionary Programming

In this algorithm, an immune operation is also accomplished with two steps, i.e. a vaccination and an immune selection, of which the former is used for raising fitness and the latter is for preventing the deterioration of population.

3.1 Vaccination

Given an individual $b \in R^n, b=(b_1,\cdots,b_n)$, a vaccination on b means modifying its some components or genes in accordance with priori knowledge so as to gain higher fitness with greater probability. The knowledge may be the value scope of some components in the optimal solution as well as the relationships among them. The vaccination should satisfy a condition, that is if b is the optimal individual, then b transforms to b with probability 1. Suppose a population is $B=(b^1,\cdots,b^N)$, then the vaccination on B suggests the operation on $n_\alpha = \alpha n$ individuals that are selected from B in proportion as α. A vaccine is abstracted form the prior knowledge of the pending problem, which the information contained in and the validity of take an important role in the performance of the algorithm.

3.2 Immune selection

This operation is accomplished by the following two steps. The first one is the immune test, i.e. testing the individuals vaccinated. If the fitness is smaller than that of the parent, which means some serious degeneration must have happened during mutation, then the corresponding parent will attend the next competition instead of the offspring individual. The second one is the annealing selection, i.e. selecting an individual d^i in the present offspring population $D_k=(d^1,\cdots,d^N)$ to join in the next parents with the probability as follows:

$$P(d^i)=\frac{f(d^i)e^{f(d^i)/T_k}}{\sum_{l=1}^{N} f(d^l)e^{f(d^l)/T_k}} \tag{5}$$

where $f(x_i)$ is the fitness of the individual d^i and $\{T_K\}$ is the temperature-controlled series tending towards 0.

3.3 Presentation of the algorithm

Without loss of generality, we re-propose the problem shown in eqn.1 as the following. Suppose $\Omega \subset R^n$ is a compact set and the fitness function $f(x)$ in Ω is continuous and greater than zero, then the goal is to search the quasi-optimal solution $x^* \in \Omega$ and to make the distance between it and the theoretical optimum smaller than what is pre-given. The algorithm is as follows:

1. Initialization: First, set the precision of solutions in accordance with the demand. Due to the calculating error, we cannot ask the quasi-optimal solution x^* is equal to the theoretical optimum. Here the precision Δ is supposed to be 10^{-l}. Then, create randomly N individuals in Ω. Each component of an individual is a decimal with l bits, and all of these individuals construct the initial population A_0.

2. Abstract vaccines according to the prior knowledge or local characteristic information.

3. Calculate the fitness of all individuals in the present population A_k. If the halt conditions are satisfied, then output the results; else, continues.

4. Perform mutation on the present parent A_k and obtain the results B_k. Suppose $A_k=(a^1,\cdots,a^N)$, $a^i \in \Omega \subset R^n$, $a^i=(a_1^i,\cdots,a_n^i)$, i=1,$\cdots$, N. The mutation on a^i means to generate the new individual b^i in the following fashion:

$$b_j^i = a_j^i + \xi_j^i, \quad j=1,\cdots,n \tag{6}$$

where ξ_j^i is the greatest decimal with l bits no more than the stochastic variable $\eta_j^i \sim N(0,\sigma_{ij}^2)$, and the variance σ_{ij}^2 is calculated in accordance with:

$$\sigma_{ij}=\frac{T_k M_j + m_j}{T_k + f(a^i)} \tag{7}$$

where T_k is the series as same as that in the annealing selection, and T_0 should be enough great so as to ensure all individuals have more evolutionary chance in the initial stages. M_j and m_j relate to the shape, size and the other factors of Ω, and $m_j \neq 0$.

5. Perform vaccination on B_k and obtain C_k;

6. Perform immune selection on C_k and obtain the next parent A_{k+1}, then go to 2.

4. The Immune Evolutionary Strategy

1. Set the precision of solutions according to the demand. We cannot regard the quasi-optimal solution x^* as the theoretical optimum because of calculation errors. The error precision Δ is supposed to be 10^{-l}.

2. Generate randomly μ individuals as the initial parent population. After the first implementation, the searching space of the quasi-optimal solution x^* is then fixed which we set as X, namely the space X is composed

with the mesh points in R^n, each component of which is integer times larger than $\Delta = 10^{-l}$. An individual in the evolutionary population means a point (x,σ) in which $x = (x_1, x_2, \cdots, x_n) \in X \subset R^n$ means an approximate solution to the pending problem, each component of which is integer times larger than Δ. The goal is to seek $x^* \in X$ ensuring $f(x^*) = \max_{x \in X} f(x) =: f^*$ during the evolution of x. Here, we let if $x \notin \Omega$, then $f(x) = 0$. For the effectual optimization of the algorithm in X, x in the initial population is generated according to uniform distribution in Ω. $\sigma = (\sigma_1, \sigma_2, \cdots, \sigma_n) \in R_+^n$ in an individual suggests the direction and step-length in the next searching. σ_i in the initial population is generated independently as follows:

$$\sigma_i = e^{u_i}, \quad u_i \sim U([\alpha_i, \beta_i]) \quad i = 1, 2, \cdots, n, \tag{8}$$

where $u_i \sim U([\alpha_i, \beta_i])$ means u_i submits to an uniform distribution on $[\alpha_i, \beta_i]$ associating with the size of Ω.

3. Crossover: generate a middle population with the size of 2μ. For each individual (x,σ) in the parent population, another individual (x^1, σ^1) is selected from there. Two new individuals are produced by the crossover of (x,σ) and (x^1, σ^1). The crossover of x and x^1 on a cross-site can result in two solutions, y and y^1, which have the same searching step-length $\sigma' = (\sigma_1', \sigma_2', \cdots, \sigma_n')$,

$$\sigma_i' = u_i \sigma_i + (1 - u_i)\sigma_i^1, \; u_i \sim U([0,1]) \quad i = 1, \cdots, n. \tag{9}$$

(y, σ') and (y^1, σ') are just the two individuals of the middle generation through the crossover of (x,σ).

4. Mutation: the mutation on an individual (x,σ) generates a new one (x', σ'):

$$\sigma_i' = \sigma_i e^{u_i}, \quad x_i' = x_i + \xi_i, \quad i = 1, 2, \cdots, n, \tag{10}$$

where $u_i \sim U([\alpha_i, \beta_i])$, ξ_i is the greatest decimal with l bits not more than the random variable $\eta_i \sim N(0, \sigma_i'^2)$.

5. Vaccination: suppose an individual is $(x,\sigma) \in R^n \times R^n$, the vaccination on (x,σ) suggests modifying some components of x and σ according to the prior knowledge. This kind of knowledge may be either rough range of some components of the optimal individual, or conditions among some components. The vaccination should ensure that if $f(x) = f^*$, then (x,σ) transfers to itself with probability 1. The vaccination on a population with size of 2μ means an operation on μ individuals selected randomly from the current population with equipotent probability.

6. Immune selection: the immune selection includes two steps, i.e. the immune test and the selection. The former means to have a test on an individual received a vaccine, if the fitness is smaller than that of the parent, which indicates that serious degeneration must have happened in the process of crossover or mutation, then the parent may attend the next competition instead of the individual. The selection suggests selecting the first μ individuals from the current population with the size of 2μ to compose a new parent population.

7. Halt test: if the halt conditions are not satisfied, then let k=k+1 and turn to step 3.

5. Constructing Vaccines

5.1 Mechanism of an immune operator

In these algorithms, each immune operator of them utilizes vaccines to intervene the variation of genes in individual chromosome. During the actually operational process, a detail analysis is firstly carried out on the pending problem, and at the same time, the basic characteristics of the problem ought to be found as many as possible. Then, the characteristics are abstracted to be a schema under some conditions for solving the problems. Finally, the schema is used as the basis of the immune operator to generate new individuals. Here it is necessary to point out that there is usually not only one characteristic in a certain problem, which also means there may be not only one vaccine which could be abstracted. Therefore, the injection can be carried out through either selecting any vaccine randomly or getting them together according to a certain logic relationship. On the other hand, a vaccine can be regarded as estimation on the schemata that the optimal individual x_{max} may match, and the accuracy of this estimation depends on further test in the later immune selection. The accuracy and the quality of a vaccine only affect the functions of vaccination, and would not influence the convergence of the algorithm.

Theorem 5.1: Under the immune selection, if the vaccination makes the fitness of an individual vaccinated higher than the average fitness of the current population, then the schema to which the vaccine is corresponding will be diffused at an index level in the population. Else, it will be restrained or be attenuated at the index level.

5.2 Methods of selecting vaccines

Evolutionary algorithms are suitable for hard problems or non-complete polynomial (NP) problems which are apt to be solved or to be found the solving rules under some local conditions when their sizes get smaller. In view of these problems, we can find some solving rules not only according to the characteristic information of them but also through decreasing the size or increasing some local conditions to simplify them. These rules simplified can be just regarded as an approach to abstract vaccines. It is necessary to consider the following two situations. More completely the original problem is decomposed locally, more evidently the solving rules will be gained, however the calculation load for obtain them will increase greatly. On the other hand, a vaccine is used for gaining the optimal solution with some local information, i.e. estimating the scheme of some genes in the optimal individual, so vaccines needn't to be very accurate. Therefore, we can select an existing algorithm with optimal iteration for abstracting them according to the practical problem decomposed locally.

Taking TSP for example, we can regard the shortest distances between every two cities among all members as a kind of vaccines through analysis. In accordance with this local characteristic information. We may select EP as the method for abstracting vaccines which is shown in the following. For the purpose of convenient expression, we first give some specific symbols. Suppose *random*(1, *n*) is a random positive integer from 1 to *n*, *Floor*(*x*) means the maximal positive integer not more than *x*, *Gauss*(*x*, *m*) indicates a normal distribution with mean *x* and variance *m*.

Algorithm 5.1 An **EP** of abstracting vaccines

Begin:

Initializing the sequence of neighboring cities:

$Neighbor(i) = random(1, n) \quad i = 1, \cdots, n$;

Initializing the shortest sub-paths:

$Sub\_path(i) \quad i = 1, \cdots, n$;

While (Conditions = True)

For $i = 1$ to n

Mutation:

$Neighbor(i) = Floor(Gauss(Neighbor(i), m))$;

Selection:

If $Distance(City\_i, Neighbor(i)) < Min\_distance(i)$

Then $Sub\_path(i) = Neighbor(i)$;

$Min\_distance(i) = Distance(City\_i, Neighbor(i))$;

End

However, in most cases of dealing with some problems, it is difficult to abstract the characteristic information of them because of lacking the mature priori knowledge. On the other hand, the work of searching the local scheme used for global solution makes the workload increase greatly and the efficiency decrease, therefore the value of this work is lost. At this time, we can abstract information form genes of the present optimal individual to make vaccines during evolutionary process. The whole algorithm with self-adaptive abstracting vaccines is expressed as follows:

Algorithm 5.2 IEP with abstracting vaccines adaptively

Begin:

$k = 0$;

While (Conditions = True)

$a_k^{optimal} = Statistics(a_k^i \mid i = 1, \cdots, n)$;

$H = \{h_j = a_{k,j}^{optimal} \mid j = 1, 2, \cdots, m\}$;

Gauss mutation: $a_k^i = Mutation(a_k^i), i = 1, \cdots, n$;

For $i = 1$ to n

If $\{P_V\}$=True

$J = random(m)$;

Vaccination: $a_{H,k}^i = Vaccine(a_k^i, h_J)$;

Immune test: If $a_{H,k}^i < a_{k-1}^i$, then $a_k^i = a_{k-1}^i$;

Else $a_k^i = a_{H,k}^i$;

Annealing selection: $A_{k+1} = S(A_k)$;

$k = k+1$;

End

where $a_{H,k}^i$ is a middle individual through vaccinating the *i*-th individual a_k^i of the *k*-th generation, P_V is the probability of vaccination, $Vaccine(a_k^i, h_j)$ suggests the operation of altering some genes of an individual a_k^i according to the schema h_j. *n* and *m* are the size of the population and the number of genes respectively, *Random*(*m*) means to generate randomly an arbitrary positive integer between 1 and *m*.

6. Simulations

6.1 Solution to TSP with IGA and IEP

First we can realize the performance of these immune evolutionary algorithms with examples of TSP. TSP is a typical NP-hard problem which is difficult to gain a solution with normal algorithms. To some extent, it is considered as an indirect criterion among the intelligent algorithms[16],[17]. Based on this consideration, we observe and study the properties of IGA with an example of 75-city TSP. For the convenience of expression, a permutation of the order to visit these *n* cities is adopted as the coding of TSP, i.e.,

$$X_i = x_i^1 - x_i^2 - \cdots - x_i^j - \cdots - x_i^n,$$

where x_i^j means the *j*-th component of an individual X_i. The fitness is calculated according to the following equation:

$$f(X_i) = \frac{76.5 \times L \times \sqrt{n}}{D_{X_i}} \tag{11}$$

where *L* means the side of the smallest square which can contain all the cities.

In this simulation, the two-point crossover is used in principle, whose positions are selected randomly (so one-point crossover may occur during the actual implementation). A novel method of only altering partial path is adopted in mutation, which is based on evaluating the inheritance to the characteristic of genotypes in the genetic individuals and the diversification of characteristics necessary for further evolution. One part of the entire path is selected every time, of which the beginning and the end are defined in accordance with the results of evaluation. The mode of exchanging for *n* times is adopted in actual operation, in which *n* is calculated as follows:

$$n = \left[N/M + \exp(-\alpha K) \right] \tag{12}$$

where N is the number of the cities, M means the number of sub-paths, K suggests the number of generations and α is a constant denoting the variation of n with K.

In the immune operator, we regard the shortest distances between every two cities among all members as the vaccines. After every genetic operation, we select some individuals for vaccination in accordance with the immune probability, after which the immune test is to be continued, i.e. testing the individual vaccinated as follows. If the fitness rises, then next operation; else make the parent take part in the competition of selection instead of the offspring. During the selection, the probability of the individual in offspring to be selected is calculated according to eqn.2.

In the actual test, the annealing temperature T in the immune selection is calculated as follows

$$T_k = \ln(\frac{T_0}{k} + 1), \quad T_0 = 100. \tag{13}$$

where k is the evolutionary generations. With the basic parameters fixed, TSP is solved with GA which maintains the best individual found over time after selection and IGA respectively, between which a comparison about the variation of the fitness with generations is shown as fig.1. IEA finds the finally best solution as same as that of Fogel[16] with the length 549.180 and the fitness 96.5 after 960 generations while GA does after 3550. All the vaccines used in IGA are shown as Fig.2.

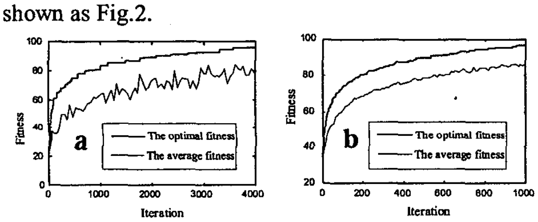


Fig.1 Comparison between canonical GA and IGA about the variation of the fitness with iteration, a. with canonical GA, b. with IGA

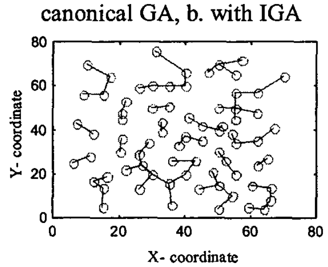


Fig.2 Vaccines used for the 75-city TSP

We can also study the properties of IEP with other example of TSP, *Grötschel's* 442-city TSP. In this simulation, methods of coding and calculating fitness are similar to the above. However in the mutation, an integer Δx_i^j is first generated with Gauss distribution $N(0, 145^2)$ whose absolute value is not greater than the number of cities. Then, the point is moved forwardly or back-forwardly for $|\Delta x_i^j|$ bits in accordance with the positive or negative sign of the integer(when overflow, the point is moved for $j + |\Delta x_i^j| - 442$ from the head or foot of code). Finally, the city pointed is exchanged with x_i^{j+1}.

In immune operation, vaccines are firstly abstracted with the algorithm 5.2. Then m genes (m is a random integer between 0 and 44) are modified according to vaccines. The annealing temperature T is calculated as eqn.13.

With the basic parameters fixed, the TSP is solved with EP and IEP respectively, between which a comparison about the variation of the optimal and the average fitness among individuals in offspring with generations is shown as fig.3 (the length of the optimal path is 5154.1 and the corresponding fitness is 124.8187, and it can be seen from this figure that EP finds the optimal solution at about the 28000 generation while IEP at 3900). During searching the solution with IEP, all the vaccines selected and the final optimal path are shown respectively as (a) and (b) in fig.4. From fig.3, it can be found out that IEP are conducive to raise the searching efficiency and restrain degradation during the evolution in original EP, so can increase the convergent speed greatly.

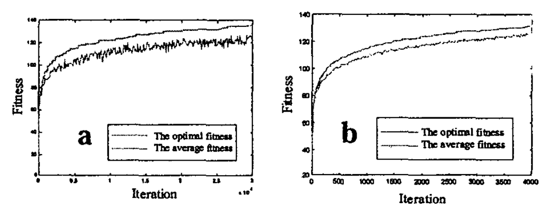


Fig.3 Comparison between canonical EP and IEP about the variation of the optimal and the average fitness among individuals in offspring with generations, a. with EP, b. with IEP

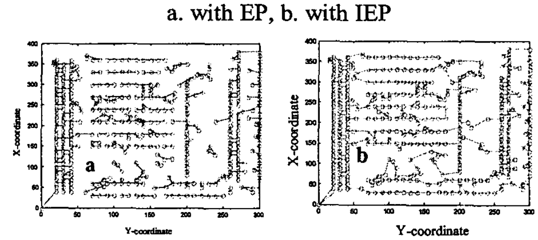


Fig.4 A solution to *Grötschel*'s 442-city TSP, a. vaccines(with iteration for 70 times), b. the optimal path

6.2 Application in function optimization

Following the above, we may also examine the searching ability of IES with an example of function optimization as follows:

$$f(x) = 10 + \frac{\sin\left(\frac{1}{x}\right)}{(x-0.16)^2 + 0.1}, \tag{14}$$

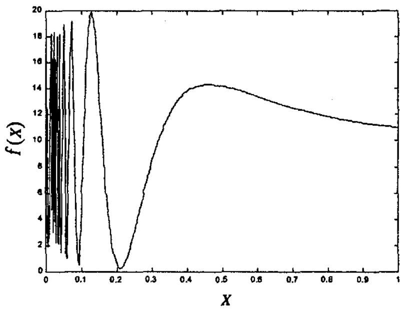


Fig.6 The function curve on (0,1]

where the variation of the function with the independent variable x in (0,1) is shown in Fig.5. Our purpose is to search x_{max} in (0,1) which satisfies the following inequality:

$$f(x_{max}) \geq f(x), \forall x \in (0,1). \tag{15}$$

For the convenience of comparison between ES and IES, we adopt an unified coding for the problem as follows:

0.	x_k^1	x_k^2	x_k^3	x_k^4

where $x_k^j (j = 1,2,3,4)$ is an integer in [0,9] which expresses the j-th gene of the k-th individual.

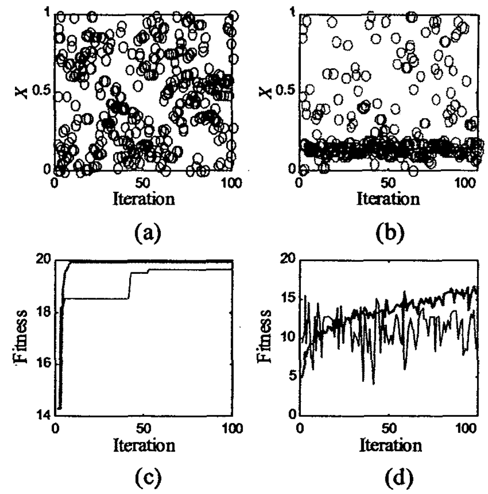
In the actual test, we let the size of population is 5 for convenient observation, the fitness function is just the original function and the halt condition is the maximum iteration (i.e. 100). In the first group, we generate randomly 20 initial populations, test each by each and observe in the following four ways:

1. The distribution of the individuals during evolution with no vaccination, i.e. the process of ES.
2. The distribution of the individuals during evolution with vaccination, i.e. the process of IES and the probability of vaccination is 0.5.
3. The variation of the optimal fitness with the reproduction of offspring.
4. The variation of the average fitness with the reproduction of offspring.

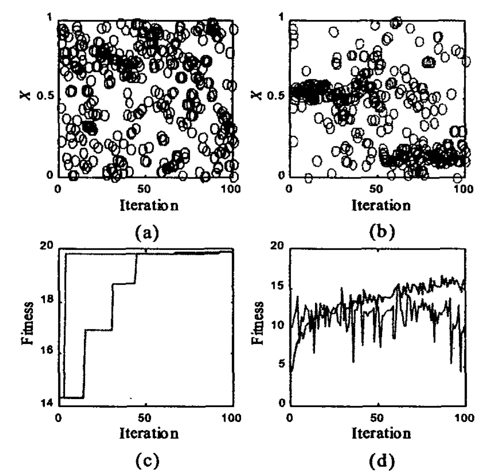
The results for 20 times vary in detail in the above ways, however the whole tendency is identical. One of them is shown in Fig.6-1, where IES finds the globe optimum($f(x)_{max}$ =19.8949; $x_{optimal}$ =0.1275) after 12 iterations, while ES only finds the local optimum ($f(x)$=19.8903; x=0.1273) after 53 iterations.

In the second group, we generate randomly another 20 initial populations. This time, we adopt an error vaccine, i.e. $x_k^1 = 5$, and also observe in the above four ways. One of them is shown in Fig.6-2, where IES finds the globe optimum after 92 iterations, while ES only finds the local optimum ($f(x)$=19.8797; x=0.1284) after 67 iterations. From Fig.7-2(b), we can see that if the vaccine is selected incorrectly, then it may be restrained step by step and the population will tend to the schema with a high fitness.

In a word, the correct selection of a vaccine takes a great important role in executing the algorithm, however how it is selected would not affect the convergence. On the other hand, the immune selection can restrain the schema with low fitness and strengthen the high one, so it is the base of ensuring the algorithm convergent.



7-1 Accepting self-adaptive vaccine



7-2 Accepting an wrong vaccine $x_k^1 = 5$

Fig. 7 Curves of the evolutionary process based on (μ,2μ)-ES and (μ,2μ)-IES, a. distribution of individuals during the evolutionary process of (μ,2μ)-ES, b. distribution of individuals during the evolutionary process of (μ,2μ)-IES, c. comparison between (μ,2μ)-ES and (μ,2μ)-IES on the optimal fitness, d. comparison between (μ,2μ)-ES and (μ,2μ)-

IES on the average fitness;
—— curve of $(\mu,2\mu)$-ES; —— curve of $(\mu,2\mu)$-IES.

7. Summary

Three novel global parallel algorithms, IGA, IEP and IES are presented, which get the immune mechanism and the evolutionary mechanism together. The strategies and methods of selecting vaccines and constructing an immune operator are also proposed in this paper. The analysis in theory and simulations show that these algorithms are not only feasible but also effective, and they are conducive to alleviate the undulate phenomenon in the existent evolutionary algorithms, therefore improve their performance and increase the convergent speed greatly.

References

1. Goldberg D. E., *Genetic algorithms in search, optimization and machine learning,* Addison-Wesley, Reading, MA,1989.
2. Holland J.H. and Reitman J. S., *Cognitive systems based on adaptive algorithms,* in Waterman D. A. and Hayes-Roth F.(Editors), *Pattern-Directed Inference Systems,* Academic Press, New York, 1978.
3. Fogel L. J., Owens A. J. and Walsh M. J. *Artificial Intelligence Through Simulated Evolution.* John Wiley, Chichester, UK, 1966.
4. Schwefel H.P and Back T. Evolution Strategies I & II. In: Winter G., Periaux J., Galan M. and Cuesta P. eds. Genetic Algorithms in Engineering and Computer Science. New York: Wiley, 1995, 111~140.
5. Muhlenbein H., *Parallel genetic algorithms in combinatorial optimization,* in Balci O., Sharda R. and Zenios S. A.(Editors), *Computer Science and Operation Research —New Developments in Their Interfaces,* Pergamon Press, New York, 1992, pp.441-453.
6. Grefenstette J. J., Gopal R., Rosmaita B., and Van Gucht D., *Genetic algorithms for the traveling salesman problem,* in Grefenstette J. J.(Editor), *Proceedings of the First International Conference on Genetic Algorithms and Their applications,* Lawrence Earlbaum Associate, Hillsdale, NJ, 1985: pp.160-168.
7. Miller G., Todd P., and Hedge S., *Designing neural networks using genetic algorithm,* in Schaffer D. J.(Editor), *Proceedings of the Third International Conference on Genetic Algorithms and Their applications,* Morgan Kaufmann, 1989: pp.360-369.
8. Riolo R. L., *Modeling simple human category learning with a classifier system,* in Belew R. K. and Booker L. B.(Editor), *Proceedings of the Fourth International Conference on Genetic Algorithms and Their applications,* San Mateo, CA, Morgan Kaufmann, 1991: pp.324-333.
9. Tokinaga S. and Whinston A. B., *Applying adaptive credit assignment algorithms for the learning classifier system based upon the genetic algorithm, IEICE Trans. on Fund. Elec. Comm. and Sci.,* Vol.E75A, No.5, 1992: pp.568-577.
10. Koza J.R., *Genetic Programming,* Cambridge, MA: MIT Press, 1992.
11. Kristinsson K. and Dumont G. A., *System identification and control using genetic algorithms, IEEE Trans. on Sys., Man and Cybernetic,* Vol.22, No.5, 1992: pp.1033-1046.
12. Krishnakumar K. and Goldberg D. E., *Control system optimization using genetic algorithms, Journal of Guidance, Control and Dynamics,* Vol.15, No.3, 1992: pp.735-740.
13. Chen R.(Editor) *The Bases of Immunology.* The Peoples Health Press, Peking, 1980: pp.54-56.
14. Zhang J. S. Xu Z. B. and Liang Y. *The whole annealing genetic algorithms and their sufficient and necessary conditions of convergence.* Science in China(Series E), Vol27, No.2, 1997: pp.154-164.
15. Rudolph G. *Convergence analysis of canonical genetic algorithms. IEEE Transactions on neural networks.* Vol.5, No.1, 1994: p96-101.
16. Fogel D. B. *Applying evolutionary programming to selected traveling salesman problems.* Cybernetics and Systems, 1993, 24, pp.27-36.
17. Lawer E. L., Lenstra J. K., Rinnooy Kan A. H. G. and Shmoys D. B. *The Traveling Salesman Problem,* John Wiley, Chichester, UK, 1985.
18. Fogel D. B. *Asymptotic convergence properties of genetic algorithm and evolutionary programming: Analysis and experiments.* Cybernetics and Systems, Vol.25, No.3, 1994, pp.389-407.
19. Fogel D. B. *Evolutionary Computation.* New York, IEEE Press, 1995, pp.135-155.